

Webes alkalmazások fejlesztése

1 beadandó dokumentáció

Elektronikai webáruház

Csernák Tamás, PAIOW2

tamas.csernak91@gmail.com

Feladat leírása:

Készítsük el egy elektronikus termékekkel foglalkozó cég online rendszerét, amelyben a vevők megrendelhetik termékeiket. A webes felületen a vásárlók adhatnak le online rendeléseket a cégnek.

- A főoldalon kategóriánként (pl. szórakoztató elektronika, számítástechnika) jelennek meg a termékek (a kategóriát kiválasztva listázódnak). Egy oldalon legfeljebb 20 termék jelenik meg, a többi lapozással lehet elérni.
- A termékek gyártóval, típusszámmal, rövid leírással, valamint nettó/bruttó árral rendelkeznek. A listát ár, illetve gyártó szerint rendezhetjük (növekvő/csökkenő) sorrendbe.
- A vásárló tetszőleges számban helyezheti a kosárba a termékeket (egy termékből több darabot is rendelhet), majd megadhatja adatait (név, cím, telefonszám, e-mail cím), és véglegesítheti a rendelést. A véglegesítés előtt a kosárból törölhetünk is, illetve változtathatjuk a darabszámot (amennyiben ez 0-ra csökken, akkor a termék törlődik), illetve az egész kosarat kiüríthetjük egy lépésben. A kosarat bármikor megtekinthetjük, illetve láthatjuk az végösszeget is.

Feladat elemzése:

Az oldal két fő részből (controllerből) fog állni:

1. Home: ez a termékek megjelenítéséért felelős. Kategóriák közül lehet választani, és rendezni lehet a termékeket. Darabszám kiválasztása után pedig a kosárba lehet rakni az adott terméket.
2. Order: ez a kosár megjelenítéséért felelős. Két részből áll, felül a kosár tartalma látható táblázatos formátumban, ahol a kosárban lévő termékek számát lehet módosítani, a táblázat alatt pedig egy kitöltendő form van, amivel a rendelést lehet véglegesíteni.

HomeController osztály:

Egyetlen *Index* action-el rendelkezik, amely megkapja a kategória azonosítóját, a rendezés paraméterét, és az oldalszámot. Létrehozza a paramétereknek megfelelően a termékek listáját, rendezi azt, majd visszatér egy nézettel ami mögött a terméklista van. A termékek lapozható megjelenítéséhez a *PagedList* osztályt használtam. A *toPagedList()* függvénnnyel bármilyen felsoroló adattípus konvertálható, két paraméter

megadásával pedig állítható hogy hány termék legyen egy oldalon, és hogy melyik az aktuális oldal. A rendezést az *OrderBy()* és *OrderByAscending()* függvényekkel lehet megvalósítani egy felsorolóosztályon. A rendezés állapotok karbantartására a ViewBag-et használtam.

OrderController osztály:

Két *Index()* akciója van, az egyik a Get, a másik a Post requesteket szolgálja ki.

A Get-et kiszolgáló továbbít a kosarat megjelenítő (*Cart*) akcióra, ami a sessionben tárolt ShoppingCart osztály segítségével egy terméklistát állít össze, és jeleníti meg.

A ***Cart*** view mögött lévő modell egy **Tuple<ProductList, CheckoutModel>**.

Az *AddToCart()* akciót a Home Index nézetéből hívjuk meg, mikor egy terméket a kosárba rakunk.

A *ChangeQuantity()* akciót a kosár nézet "Megváltoztat" gomb nyomásával hívjuk meg.

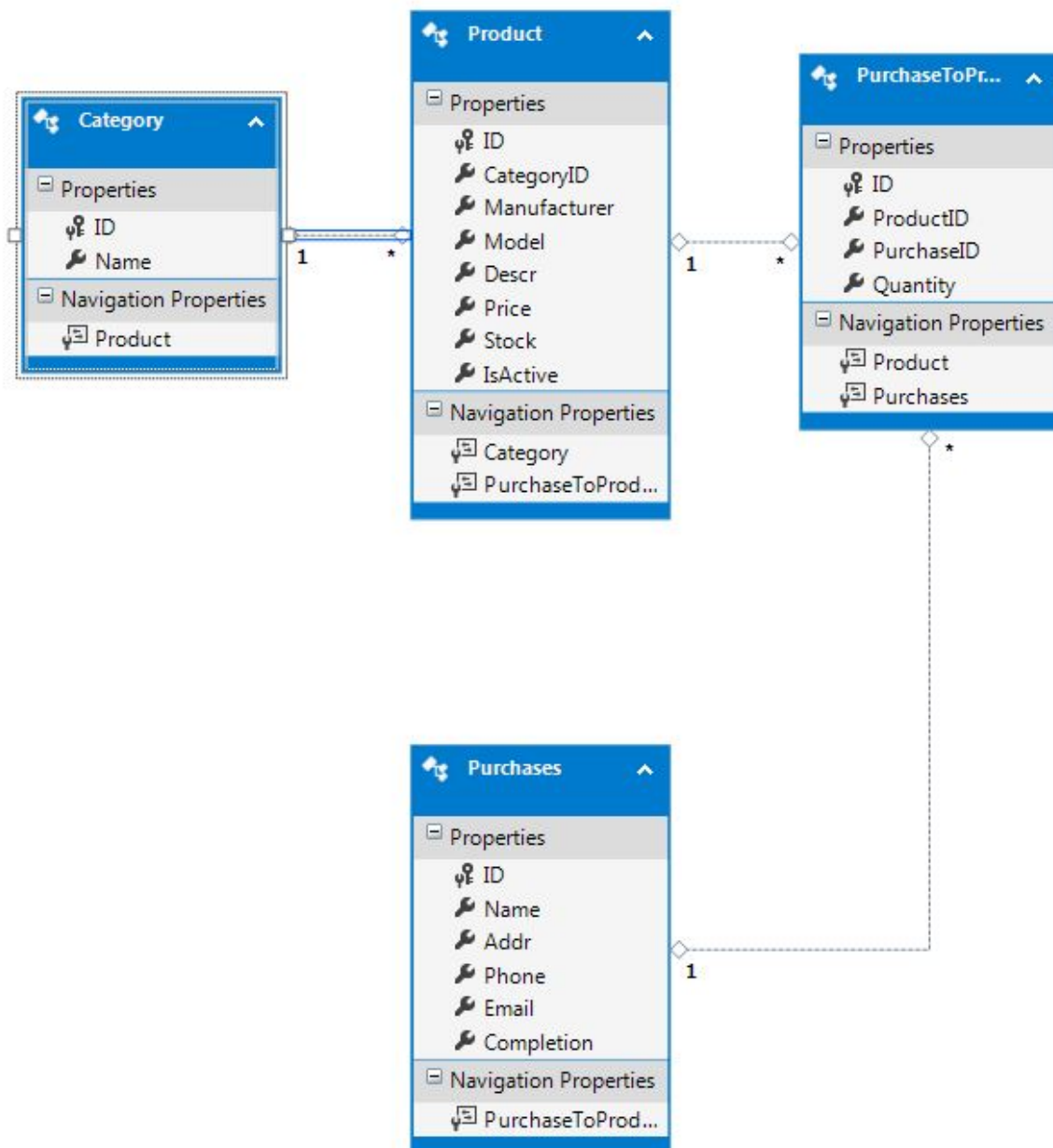
A *ClearCart()* akciót a "Kosár ürítése" gomb nyomásával hívjuk meg.

A Post-ot kiszolgáló *Index()* action végzi a rendelés végrehajtásának a logikáját.

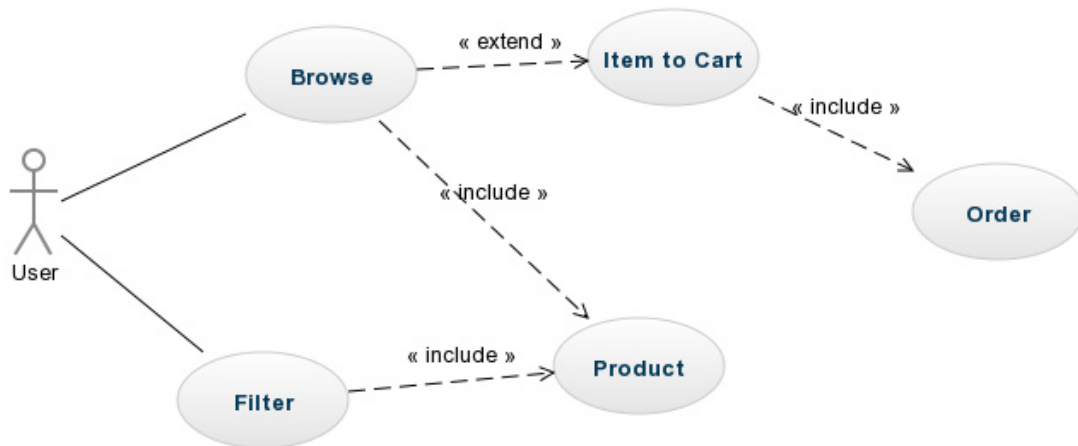
Megkapja egy CheckoutModel típusú argumentumban a vásárló adatait.

Létrehozza a kosrában lévő termékek listáját, ellenőrzi az adatokat, a termékek darabszámát, és végrehajtja a az adatbázis frissítését. Ha a rendelés sikeres, továbbít az OrderComplete nézethez, amiben egy üzenetet írunk ki a vásárlónak.

Az adatbázis felépítése a következő:



A felhasználói esetek:



A nézetek elrendezése:

- _ViewStart.cshtml
- Shared
 - _Layout.cshtml - a közös template
- Home
 - Index - a termékek megjelenítése
- Order
 - Cart - a kosár és a rendelés véglegesítéséhez a form
 - OrderComplete - a sikeres rendelésről értesítő oldal
 - Error - a hibáról értesítő oldal